Figure 3, the Abstract, and column 5, lines 56-63. Because of the unusual syntax of the rejection it is not entirely clear, but it appears to be the Examiner's position that XML/RDF in some way is a teaching of SQL.

With this in mind, let's see what Sarkar actually teaches. Of significance is that Sarkar is directed in the main to ***systems that do not have middleware at all***:

> "In one embodiment of the invention, a virtual unified Database over multiple object relational databases over the web is described. Application business logic, messaging services and object request brokers reside inside an object relational database server ***eliminating the need for a middle tier***....Therefor a uniform paradigm for multi-tier client/server ***without a middle tier application server is presented***", col. 6, lines 7-29 (emphasis mine).

And again:

> "Recent development of universal servers and object relational database servers enhances the capability of relational database servers by putting application business logic, messaging services and object brokers inside the relational database server and thereby ***eliminating the middle tier. This invention is based on a similar notion***", col.10, lines 38-44 (emphasis mine).

With the above in mind, it is clear from Sarkar that it is principally directed to systems without any middleware at all, pushing the middleware function down to the database. This is important to understand, because while it is true that Sarkar mentions middleware and SQL it is more or less in passing, without any deep or hidden teachings or suggestions about these topics other than what Sarkar briefly touches on.

And here is the bulk of what Sarkar has to say regarding middleware in the Detailed Description:

> "FIG. 3 shows multi-tier architecture for web applications. First tier in this architecture is a browser on the client site. A client needs very little software on the client computer. Any command or request for a service goes to a web server for internet services as shown in the

> figure. Internet services are usually dealing with HTTP (hypertext transfer protocol) to communicate with various web sites. Web servers talk to application servers where business-specific application logic is maintained. A fat client may talk directly to an application server. Request to application servers is made through a messaging middleware comprising of XML/RDF integration services and Common Object Request Broker Architecture (CORBA) compliant services. Two disparate application servers can also communicate through middleware utilizing CORBA and XML/RDF integration services. Hyper Text Transfer Protocol and the internet are synchronous by nature. When certain application server contacts another using HTTP there is a direct link established that is not broken until all information has been transferred. This may not be the situation for other protocols and messaging services. Object request brokers (ORB) create a persistent link between web server and application server for servicing object requests by the use of IIOP (Internet Inter ORB Protocol). Application server talks through either XML/RDF integration services or CORBA services to communicate with third tier databases for executing SQL queries and other database services as shown in FIG. 3", col. 10, lines 10-37.

Thus, in relevant part Sarkar teaches that its middleware includes XML/RDF integration services and Common Object Request Broker Architecture (CORBA) compliant services. It further teaches that object request brokers (ORB) create a persistent link between web server and application server for servicing object requests by the use of IIOP (Internet Inter ORB Protocol). Sarkar concludes this section by simply mentioning that an application server talks through either XML/RDF integration services or CORBA services to communicate with third tier databases for executing SQL queries and other database services. That's all.

With the above exposition in mind, there is no mention of a parameterized SQL statement in the middleware. All Sarkar divulges is that an application server can "talk" through XML/RDF or CORBA services in some unknown way to execute SQL queries from an unmentioned source in an undisclosed manner. Not only does this fail completely to approach a teaching or suggestion of a parameterized SQL statement, it is so bereft of detail that it arguably fails to enable the middleware system of Figure 3, a defect discussed at length in Applicant's prior response regarding another reference and incorporated herein.

Turning now to the short portion of the Summary of Sarkar that has been relied on in the rejection (col. 5, lines 56-63), nowhere is "middleware" mentioned in this section, so whatever it purports to teach, the failure to mention "middleware" more or less moots the allegation that this section relates to the claimed "middleware".

Moreover, all this section mentions about SQL is that "specifications for SQL queries", whatever they might be, "could be presented in XML/RDF documents" at some undisclosed location, and that alternatively queries "could be triggered through thin client windows" in some undisclosed way. In other words, beyond failing to mention "middleware" the relied-upon section does not even say where the XML/RDF documents are, and there is no way to know, on the evidence of record, what, exactly, the "specifications for SQL queries" are, much less that they are parameterized statements as required by, e.g., Claim 1. Indeed, nowhere else in Sarkar does the term "specifications for SQL queries" appear. Ratcheting this single bare mention of a "specification for SQL query" into the claimed "parameterized statement" is, thus, without the requisite prior art evidentiary support, and the rejection consequently cannot be sustained.

Rejections Under 35 U.S.C. §103

Claims 7, 22, and 29 have been rejected under 35 U.S.C. §103 as being obvious over Sarkar in view of Barrick, Jr. et al., with the rejection justifying the proposed combination on the ground that since Barrick, Jr. et al. teaches SOAP/GET/POST requests, it would have been obvious to drop these into Sarkar "because the HTTP GET request contains a performance parameter which can be determined the (sic) measurements of at (sic) the server (middleware) in order to send a quick response once a request is received (abstract)."

So what? This is not a prior art suggestion to combine, as is otherwise required by MPEP §2143.01. The mere fact that a reference can be modified does not render an invention obvious, unless the modification is suggested by the prior art, id., discussing In re Mills. In the present case, nothing suggests gutting the XML/RDF scheme of Sarkar and replacing it with another, much less the particular one of Barrick, Jr. et al. In fact, Sarkar teaches away from the proposed modification:

> "It is ***a primary objective*** of the present invention ***to provide a mechanism in XML and RDF***...", col. 5, second sentence of Summary (emphasis mine).

Moreover,

> "***The goal*** of system-to-system interaction ***remains primarily the domain of XML and RDF.*** Information to and from 8 is represented in XML and RDF format to represent context, meaning and relationships over data elements so that a program or an agent can operate on the data without human intervention", col. 8, starting at line 60 (emphasis mine).

Thus, not only does Sarkar fail to suggest POST/GET/SOAP, it (1) exclusively teaches only XML/RDF and (2) expressly divulges to the skilled artisan, twice, that this is a primary goal of its invention, thereby teaching away from the proposed modification, see MPEP §2142 (it is error not to give due regard to references that teach away).

Submitted,

John L. Rogitz
Registration No. 33,549
Attorney of Record
750 B Street, Suite 3120
San Diego, CA 92101
Telephone: (619) 338-8075

JLR:jg

1176-2.AM2